



# Energy Efficient Mobile Routing in Actuator and Sensor Networks with Connectivity preservation.

Essia Hamouda, Nathalie Mitton, David Simplot-Ryl

## ► To cite this version:

Essia Hamouda, Nathalie Mitton, David Simplot-Ryl. Energy Efficient Mobile Routing in Actuator and Sensor Networks with Connectivity preservation.. 10th International Conference on Ad Hoc Networks and Wireless (AdHocNow), Jul 2011, Paderborn, Germany. pp.15-28. inria-00599120

**HAL Id: inria-00599120**

**<https://inria.hal.science/inria-00599120>**

Submitted on 17 Aug 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Energy Efficient Mobile Routing in Actuator and Sensor Networks with Connectivity preservation

Essia Hamouda<sup>1,2</sup>, Nathalie Mitton<sup>2</sup> and David Simplot-Ryl<sup>2</sup>

<sup>1</sup> University of California Riverside, <sup>2</sup>INRIA Lille-Nord Europe, Univ. Lille 1, CNRS  
<sup>1</sup>essia@cs.ucr.edu, <sup>2</sup>{nathalie.mitton,david.simplot-ryl}@inria.fr

This work was partially supported by CPER Nord-Pas-de-Calais/FEDER Campus Intelligence Ambiante and the ANR BinThatThinks project.

**Abstract.** In mobile wireless sensor networks, flows sent from data collecting sensors to a sink could traverse inefficient resource expensive paths. Such paths may have several negative effects such as devices battery depletion that may cause the network to be disconnected and packets to experience arbitrary delays. This is particularly problematic in event-based sensor networks (deployed in disaster recovery missions) where flows are of great importance. In this paper, we use node mobility to improve energy consumption of computed paths. Mobility is a two-sword edge, however. Moving a node may render the network disconnected and useless. We propose CoMNet (Connectivity preservation Mobile routing protocol for actuator and sensor NETworks), a localized mechanism that modifies the network topology to support resource efficient transmissions. To the best of our knowledge, CoMNet is the first georouting algorithm which considers controlled mobility to improve routing energy consumption while ensuring network connectivity. CoMNet is based on (i) a cost to progress metric which optimizes both sending and moving costs, (ii) the use of a connected dominating set to maintain network connectivity. CoMNet is general enough to be applied to various networks (actuator, sensor). Our simulations show that CoMNet guarantees network connectivity and is effective in achieving high delivery rates and substantial energy savings compared to traditional approaches.

**Keywords** wireless communication; performance optimization; node mobility; connected dominating set.

## 1 Introduction

Wireless sensor networks are intended to be deployed in hostile environments (battlefield, forest, etc.). Therefore, it is expected that a large number of cheap simple sensor devices will be randomly scattered over a region of interest. These devices are powered by batteries and have limited processing and memory capabilities. Among numerous challenges faced while designing WSN protocols, maintaining connectivity and maximizing the network lifetime stand out as critical considerations. The connectivity condition is generally met by deploying

dense homogeneous networks to increase resources per unit area. However, dense networks can have several problems, such as device management and increased transmission interference and contention. Another approach is to use specialized nodes with long-range communication capabilities to maintain a connected network. The second consideration, network lifetime, is directly related to how long the power resources in sensor nodes will last. The network lifetime can be increased by designing and using energy efficient-protocols and algorithms. An example would be a scheduling scheme to make sensors work in batches to extend the network life [9]. Another solution is to add actuators, *i.e.* mobile sensor nodes, that can be moved to areas where resources are most needed to efficiently route packets. Actually, it has been shown [17] that deploying resource rich mobile devices in a network can provide the same performances as increasing the network density.

The main motivation of this work is to take advantage of node mobility to extend the life of the network resource and consequently the network itself. Currently, even though the idea of using mobile actuator nodes to improve the performance of the network is well recognized, there is not much work that takes advantage of node mobility to improve routing in wireless networks while ensuring network connectivity. Available solutions adopt existing routing protocols to find an initial route, and iteratively move each node to an arbitrary location on the straight line connecting the source-destination pair. However, adopted node relocation strategies may cause useless zig-zag movements of nodes [10] and may disconnect the network [12]. Moreover, in [10, 12] we argue that the associated energy optimization model is incomplete as it does not incorporate the mobility cost in the routing decision.

In this paper, we propose CoMNet, a Connectivity preservation Mobile routing protocol for actuator and sensor NETWORKS, an energy efficient position-based routing protocol which takes advantage of node mobility to minimize energy consumption and ensure network connectivity. CoMNet is based on a cost-over-progress metric where the optimized cost includes both moving and transmission costs. CoMNet has three variants, *ORouting on the move*,  $Move_{(r)}$  and  $Move_{(DSr)}$  described in more details in Section 4. These variants have the same objective but are intended for different applications. Both  $Move_{(DSr)}$  and *ORouting on the move* are intended for use in high traffic networks. They assume that the path will be highly used for a long time, so they establish a path on the straight line (the shortest distance) connecting a source-destination pair. Connecting two nodes and aligning relaying nodes on a straight line reduces energy consumption during transmission between the source and destination. Indeed, it has been shown that the straight line is the least energy consuming path connecting a source-destination pair [12]. *ORouting on the move* focuses more on optimizing the move distance while  $Move_{(DSr)}$  focuses on optimizing the transmission cost. The third variant  $Move_{(r)}$  assumes that the network traffic between the source and the destination is light thus it is not optimum to move and align nodes. All CoMNet variants have the following properties:

- *Localized*: A routing decision depends only on local information. A node has to know only its geographical location, the ones of its neighbors and of the final destination.
- *Scalable*: CoMNet is *memoryless*. No routing information needs to be stored at a node or in a message.
- *Loop free*: A message is always sent to a node in the forwarding direction of the destination.
- *Energy efficient*: In its routing decision, CoMNet considers both the minimum cost of sending a message and moving a node to its new location. At each step, it chooses the least energy consuming solution.
- *Guaranteed connectivity*: Though nodes may be mobile and may be relocated, CoMNet *guarantees* that the network connectivity is maintained at all times. This is achieved by first, relying on a connected dominating set (CDS), second, by enforcing that nodes in a CDS to be static and mobile nodes to lay within communication range of at least one dominating node.

This paper is organized as follows. Section 2 reviews literature related to CoMNet. Section 3 presents our model and assumptions. CoMNet principles along with its three variants are described in Section 4 and evaluated in Section 5. We conclude and present future work in Section 6.

## 2 Related Work

Before proceeding into the analysis, we present a brief overview of some work of the literature related to position based routing algorithms in static and mobile networks and dominating set algorithms. This is by no means exhaustive and is only indicative of the interest and the applications. For a complete survey, the reader should refer to [8].

**Position based routing in static networks.** Position based routing algorithms for static sensor networks have been widely studied in the literature. The basic principle is as follows. Every node is aware of its position, the ones of its neighbors and of the destination. A routing decision is made based only on this local information. In greedy routing [6] for instance, the node currently holding a packet forwards it to the neighbor closest to the destination. This method was then extended to energy efficient variants [11, 5], to guaranteed delivery solutions [1] and to a combination of both energy efficient and guaranteed delivery approaches [14, 4]. In *ORouting* [5], the neighbor closest to the line ( $SD$ ) connecting the source node  $S$  to the destination node is considered as the best candidate relay to minimize energy. Cost Over Progress (COP) based routing [11] is a localized metric aware greedy routing scheme. A node  $u$  forwards a packet to a neighbor  $v$  in the forwarding direction of the destination  $D$  such that the ratio of the energy consumed for sending a message from  $u$  to  $v$  (any cost metric can be used) to the progress made (measured as the reduction in distance to  $D$ ) is minimized.

**Routing in mobile networks.** Little work has been done in routing in mobile sensor networks. We will address this relationship in our analysis section.

Current solutions adopt existing routing protocols to find an initial route, and iteratively move each node to the midpoint of its upstream and downstream nodes on the route. However, these routing protocols may not be efficient. Moving strategy in [10] may cause useless zig-zag movements. In MobileCOP [12], next hop on the path is selected based on COP [11] metric. Once a path is computed, its nodes are moved and placed equidistantly on the straight line connecting the source to the destination, maintaining the same number of hops as in the computed path. Such move may induce a memory overhead on nodes, since they have to store the path, and a high transmission delay. More importantly, the network may be disconnected (a node may move out of range of its neighbors). This arbitrary move may not be optimum as when a node moves further from its upstream neighbor, the transmission delay will automatically increase. In addition, none of these approaches consider the cost of moving in the routing decision. A closer look to the basic *ORouting* (described previously) suggests that the protocol minimizes the moving cost of a candidate node to line ( $SD$ ) and locally decreases the transmission cost.

**Dominating sets.** Dominating sets (DS) are defined as follows. Each node in a graph either belongs to a dominating set or has a neighbor in the DS. The DS is called a Connected Dominating Set if the DS nodes are connected. The problem of computing the smallest CDS is known to be NP-complete even if knowledge of the global topology is available. Dai and Wu [3] introduced a generalized DS concept, where coverage can be provided by an arbitrary number of connected one-hop neighbors. The definition was modified by [15], to avoid message exchange between neighbors, as follows. Node  $a$  is covered by its one-hop neighbors  $b, c, \dots$  if these neighbors are connected. It is then further simplified in [2] as follows. First, each node checks if it is an intermediate node. Then each intermediate node  $a$  constructs a subgraph  $G'$  of its neighbors with higher key values. If  $G'$  is empty or disconnected then  $a$  belongs to the DS. If  $G'$  is connected but there exists a neighbor of  $a$  which is not a neighbor of any node in  $G'$  then  $a$  is in the CDS. If position information of 1-hop neighbors is available, nodes can decide whether or not to belong to a so defined CDS without exchanging any message with their neighbors. Note that these algorithms are local and do not incur any additional message exchange overhead.

### 3 Models

Before introducing CoMNet, we present in this section the assumptions of the protocols and define the cost models involved in its design and functionality.

**General model assumptions.** We consider a sensor network where nodes are randomly scattered, are aware of their geographical location and are able to tune their transmission range between 0 and  $R$  ( $> 0$ ). We also assume that sensor/actuator devices can be either mobile or stationary. The latter assumption can be further relaxed by making all nodes static or all nodes mobile. We denote by  $N(u)$  the set of physical neighbors of node  $u$ , *i.e.* the set of nodes in communication range of node  $u$  ( $N(u) = \{v \mid |uv| < R\}$  where  $|uv|$

is the Euclidean distance between  $u$  and  $v$ ). Let  $\delta(u) = |N(u)|$  be the cardinality of  $N(u)$ , also called the degree of node  $u$ . We also define  $N_D(u)$  the set of neighbors of node  $u$  with positive progress toward destination node  $D$ :  $N_D(u) = \{v \in N(u) \wedge |vD| < |uD|\}$ .

**Transmission cost.** We denote by  $C_s(\cdot)$  the cost measured in units of energy consumed to transmit a packet. We use the most common energy model [13]:

$$C_s(r) = \begin{cases} r^\alpha + c & \text{if } r \neq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where  $r$  is the distance between a sender and a receiver;  $c$  is the overhead (in units of energy) due to signal processing;  $\alpha$  is a real constant ( $> 1$ ) that represents the signal attenuation. The optimal transmission radius, that minimizes the total power consumption for a routing task is  $r^* = \sqrt[\alpha]{\frac{c}{\alpha-1}}$  assuming that nodes can be placed on a straight line toward the destination [16].

**Mobility cost.** We denote by  $C_m(\cdot)$  the cost, measured in units of energy consumed, to relocate a node. To the best of our knowledge, there is no accurate model to define such a cost. Therefore, in this work, we use the model adopted in the literature [12, 10]:

$$C_m(|vv'|) = a|vv'| \quad (2)$$

where  $v$  denotes a node at its original position  $(x_v, y_v)$  (before it is moved). To eliminate confusion, we refer to node  $v$  after it has moved as  $v'$  and its new position is  $(x'_v, y'_v)$ .  $|vv'|$  is the Euclidean distance between  $v$  and  $v'$  and  $a$  is a constant to be defined.

In this work, we adopt the above cost models because they are widely used and as a proof of concept. However, other cost metrics can be considered.

## 4 CoMNet

### 4.1 Motivation

The pitfall of node mobility is the risk of the network to be disconnected once nodes have moved and consequently routing fails. To support our claim, we study the behavior and performance of MobileCOP [12] in terms of network connectivity. We consider packet transmission between a single source-destination pair. As this method dictates, after every transmission, nodes along the computed paths move. Figure 4(a) displays the percentage of times the network remains connected as a function of the number of routes computed (or number of source-destination pair transmissions). Results show that for low node densities ( $\delta \leq 5$ ), after the first routing task, the network remains connected only in 75% of the cases. After computing 10 consecutive routes, the network remains connected only in 25% of the cases. In fact, node density has to reach  $\delta = 20$  to keep a network connected after 10 computed routes. As the number of computed paths increases, to keep the network connected the value of  $\delta$  has to be large.

## 4.2 CoMNet principles

To prevent the network to disconnect, CoMNet uses the notion of CDS. Furthermore, and contrary to other routing protocols, to minimize overall energy consumption CoMNet not only optimizes transmission energy but also mobility cost. In the following we describe both steps in more details.

**CoMNet and network connectivity.** We claim that as long as every node is connected and more specifically is neighbor to a node in a CDS, the network will never be disconnected. Thus, the first step in CoMNet is to locally compute a CDS, using any one of the methods described in Section 2. At this step, every node is covered by the CDS. In order to keep this property, a node is moved if and only if its final position is covered by the CDS. In CoMNet, nodes that belong to a dominating set never move during a next hop selection.

**CoMNet and routing cost.** To select a forwarding node, CoMNet uses a cost to progress criteria. Current node  $u$  chooses node  $v \in N_D(u)$  among its neighbors in the forwarding direction of the destination  $D$ . More specifically, the selected node  $v$  ( $v \in N_D(u)$ ) minimizes the ratio of the global cost (packet transmission cost and node relocation cost) to the progress made towards the destination  $D$ . Indeed,  $v$  satisfies the following optimization problem:

$$v = \underset{v \in N_D(u)}{\operatorname{argmin}} \frac{C'_s(|uv|) + C_m(|vv'|)}{|uD| - |v'D|}$$

with

$$C'_s(|uv|) = \begin{cases} C_s(|uv|) & \text{send before } v \text{ moves} \\ C_s(|uv'|) + \epsilon & \text{send after } v \text{ moves to } v' \end{cases}$$

where  $v'$  represents the position where  $v$  should move ( $v = v'$  if  $v$  is a dominating node).  $C_m$  and  $C_s$  are respectively moving and sending cost functions defined by Eq. 2 and 1. In the *send after move* case, before moving  $v$ , a beacon is sent to  $v$  to request and advertise its move to  $v'$  (the cost of sending the beacon is  $\epsilon$ ). Once  $v$  moves,  $u$  sends its message to  $v'$ . The transmission cost used depends on the CoMNet variant used which is described in the following paragraphs. The selection of the next hop is summarized in Algorithm 1.

There are major differences between CoMNet and existing solutions. First, CoMNet incorporates the cost of moving in its routing decision and it is memoryless (it does not require nodes to store a computed path to be used at a later time). Second, to preserve network connectivity in spite of node movement, in CoMNet a node moves if and only if it does not belong to the DS and its targeted position is within the communication range of at least one dominating node. Third, CoMNet is general, it can assume that all nodes are mobile or all nodes are static. It can be easily adapted to heterogeneous networks composed of both actuators and sensors, by simply setting  $v = v'$  for sensor nodes which cannot move.

Now we describe CoMNet variants, *ORouting on the Move*,  $\text{Move}_{(DSr)}$ ,  $\text{Move}_{(r)}$ . The selection principles is the same for all variants. They differ mainly in the computed cost criteria and in the node relocation scheme. We will show that each variant has its own advantages depending on the application.

---

**Algorithm 1** SelectNextHop(u,D) - Run at node  $u$  toward destination node  $D$ .

---

```

1: if  $N_D(u) = \emptyset$  then
2:   Return NULL {Routing has failed.}
3: end if
4:  $A \leftarrow \emptyset$ 
5: for all  $\{v \in N_D(u)\}$  do
6:    $v' \leftarrow \text{NewLocation}(u, v, D)$  {Compute position  $v'$ .  $v = v'$  if  $v \in CDS$ }
7:   if  $\{\exists a \in CDS \mid |av'| < R\}$  then
8:      $A \leftarrow v$ 
9:   end if
10: end for
11:  $w \leftarrow \underset{v \in A}{\text{argmin}} \frac{C_m(|vv'|) + C'_s(|uv|)}{|uD| - |v'D|}$ 
12: Return  $w$  {Routing has succeed.}

```

---

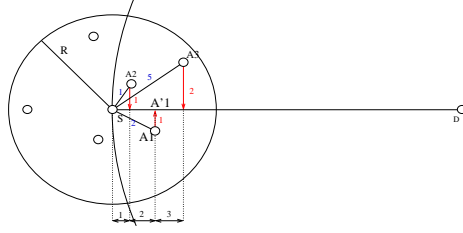
### 4.3 CoMNet: ORouting on the move

*ORouting on the move* is based on the plain ORouting [5] algorithm. It assumes that the traffic sent from the source to the destination is high and a computed path will be highly utilized for a long time. Therefore, the optimal path computed is on a straight line connecting the source to the destination. *ORouting on the move* further aligns nodes on the straight line in order to reduce the overall path length and consequently the overall routing energy consumed. Source node  $S$  selects next hop neighbor  $A$  based on the cost-to-progress criteria then takes advantage of mobility and moves the selected node. In Figure 1, which illustrates such a routing, node  $S$  has to select the next hop among nodes  $A_1, A_2$  and  $A_3$  (nodes in forward direction to  $D$ ). Cost over progress metric is computed for each node ( $\frac{2+1}{3} = 1$  for  $A_1$ ,  $\frac{1+1}{1} = 2$  for  $A_2$  and  $\frac{5+2}{6} = 1.17$  for  $A_3$ ). Having the smallest cost, node  $A_1$  is selected and moved to location  $A'_1$ , intersection point of the line connecting the source to the destination and its perpendicular line passing through  $A_1$ . Note that *ORouting on the move* objective is to minimize the move distance. *ORouting on the move* is thus well adapted to situations where moving is very costly. In this case,  $S$  first needs to send a beacon to node  $A_1$  to request and advertise its move on the line. Once  $A_1$  moves to position  $A'$ ,  $S$  forwards the message to  $A'$ . The sending cost is  $C'_s(|SA'|) + \epsilon$  and the move cost is  $C_m(|AA'|)$ .

### 4.4 CoMNet: Move<sub>(DSr)</sub>

As in *ORouting on the move*, *Move<sub>(DSr)</sub>* aims at aligning nodes along a straight line in order to reduce the energy consumed by consecutive transmission from  $S$  to  $D$ . The objective in this case is to move the selected neighbor to a new position  $r'$  distant from the source (or relay) node to optimize routing energy consumption. The idea here is to compute a routing path from  $S$  to  $D$  where all nodes are aligned on line  $(SD)$  and where all hop lengths are equal to the optimal transmission distance  $r^*$ . Since this objective is not always achieved ( $\frac{|SD|}{r^*}$  is not



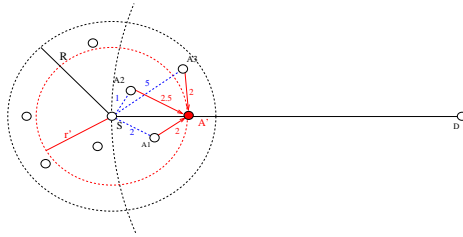


**Fig. 1.** *ORouting on the move.* Red arrows show possible displacement of nodes with associated moving costs. Solid links from  $S$  to  $A_i$  are associated to sending costs.

always an integer), we opted for a hop length  $r'$  close to the optimal value such that all the nodes in the path are equidistant. The challenge here is to determine the proper value of  $r'$ . As mentioned in Section 2, since the optimal range  $r^*$  has a closed form expression, the optimal number of hops  $n^*$  is expressed as  $n^* = \left\lceil \frac{|SD|}{r^*} \right\rceil$ . We can thus infer that  $r'$  is equal to

$$r' = \begin{cases} \frac{|SD|}{n^*} & \text{if } r^* - \frac{|SD|}{n^*} < \frac{|SD|}{n^*-1} - r^* \\ \frac{|SD|}{n^*-1} & \text{otherwise} \end{cases} \quad (3)$$

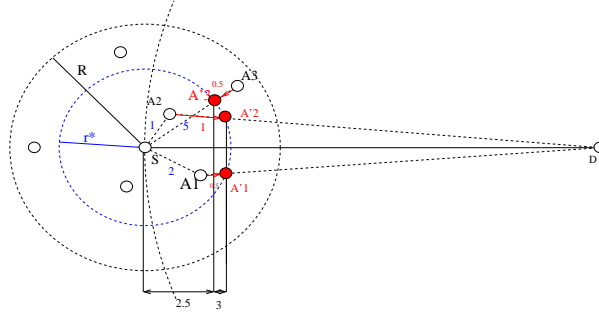
Note that in the rare case where  $|SD| = kr^*$ ,  $k$  a positive integer value,  $r' = r^*$ . An illustration of this method is given in Figure 2 where nodes  $A_i, i \in \{1, 2, 3\}$  are neighbors of  $S$  in the forward direction. Only one of the nodes  $A_i$  will be selected by  $S$  based on the energy cost. In the case presented in Figure 2,  $A_2$  will be selected (cost is  $2 + 2 = 4$  for  $A_1$ ,  $2.5 + 1 = 3.5$  for  $A_2$  and  $5 + 2 = 7$  for node  $A_3$ ) and will move to  $A'$  location such that  $|SA'| = r'$ . Clearly, the progress is a constant for a given neighborhood and does not affect the objective function. It is worth noting that if a node  $B$  already lays in position  $A'$ , it is useless to move  $A'$ .  $B$  will be the selected neighbor. Once the neighbor is selected the packet is transmitted before the node is moved to its new location. The sending cost here is  $C'_s(|AA_2|)$ .



**Fig. 2.** *Move<sub>(DSr)</sub>.* Red arrows show possible displacement of nodes with associated moving costs. Dashed links are associated to the sending cost.

#### 4.5 CoMNet: $\text{Move}_{(r)}$

As in  $\text{Move}_{(DSr)}$ , the objective is to select a neighbor and move it to a new position  $r^*$  distant from the source (relay) in order to decrease the sending cost. But contrary to the previous methods, the new location has to be on the circle  $\mathcal{C}_{r^*}$  of radius  $r^*$  centered at  $S$ . The main objective is to reduce the moving distance while minimizing the sending cost. This variant of CoMNet is used in light traffic network where routes are seldomly used for long transmission and do not need to be fully optimized to align nodes on line  $(SD)$ . On the other hand, if node  $A$  is such that  $|SA| < r^*$ ,  $A$  should be relocated on the intersection of  $\mathcal{C}_{r^*}$  and line  $(AD)$ . If node  $A$  is such that  $|SA| > r^*$ ,  $A$  should be relocated on the intersection of  $\mathcal{C}_{r^*}$  and line  $(SD)$ . An illustration of this method is given in Figure 3 where nodes  $A_i$ ,  $i = 1, 2, 3$  are neighbors of  $S$  in the forward direction. Only the selected neighbor  $A_i$  will be moved to node  $A'_i$  location. Node  $S$  selects the node which provides the smallest positive COP ratio, *i.e.* node  $A_2$  (COP for  $A_1$  is  $\frac{2+0.5}{3} = 0,83$ , COP for  $A_2$  is  $\frac{1+1}{3} = 0,66$ , COP for  $A_3$  is  $\frac{5+0.5}{2.5} = 2,2$ ). As in  $\text{Move}_{(DSr)}$ , the packet is transmitted before the node is moved to its new location. The sending cost is equal to  $C'_s(|AA_1|)$ .



**Fig. 3.**  $\text{Move}_{(r)}$ . Red arrows show possible displacement of nodes with associated moving costs. Dashed links are associated to the sending cost.

### 5 Analysis and experimental results

We compare the performance of all CoMNet variants *ORouting on the move*,  $\text{Move}_{(DSr)}$  and  $\text{Move}_{(r)}$  to MobileCOP. We use WSNNet/Worldsens [7] event-driven simulator for large scale wireless sensor networks, that assumes 802.11 DCF MAC layer, free space propagation model and packet collisions. For simulation purpose we assume that nodes are uniformly distributed over a  $1000 \times 1000$  square and can adapt their range between 0 and  $R = 150$ . The variants are compared for the same system environment: same samples of node distribution, same source-destination pairs for various network node densities. We consider only

connected networks. We compare the variants based on resource consumption and routing success.

### 5.1 Routing success rate

Figure 4(b) shows the percentage of routing success with respect to network node density. Results show that CoMNet outperforms MobileCOP for low densities. For higher densities CoMNet and MobileCOP have similar performance. As expected, *ORouting on the move*, which routing behavior is the closest to MobileCOP, performs very close to MobileCOP. The best performing technique is *Move<sub>(r)</sub>* variant.

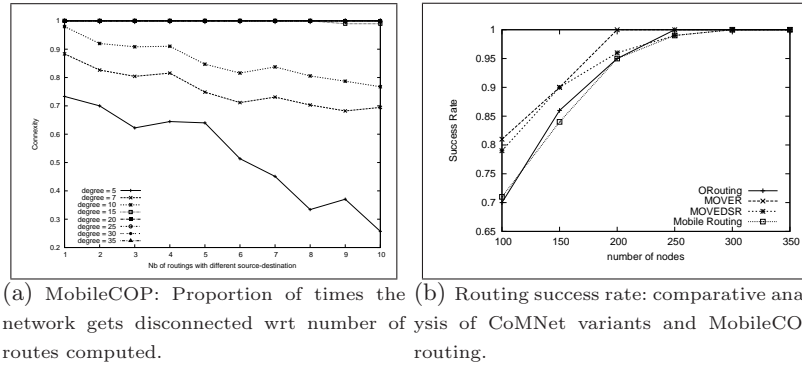


Fig. 4. Disconnection and success rate

### 5.2 Energy consumption

We evaluate the energy consumption of each algorithm based on the energy models described in Eqs. 1 and 2. To compute  $C'_s(\cdot)$  the energy consumed to send a packet, we use parameter values used in the literature [11], *i.e.*  $c = 3^8$  and  $\alpha = 4$ , which lead to an optimal transmission range of  $r^* = 100$ . As stated previously, nodes will be equipped with GPS devices for localization purpose and we assume that the energy consumed for each node to identify its location is the same for all nodes. We also assume that the cost to exchange *Hello* packets is the same for every node. As such, these consumptions are considered as a constant and will not be included in our energy optimization model. For each source-destination pair our simulator computes 10 consecutive routes between a given source-destination pair and averages the computed statistics of more than 1000 tries. Regarding the mobility model's (Eq. 2) parameter  $a$ , since not much research has been done in this area, we run simulations for three different values of constant  $a$ , computed as follows.

1. if sending is as costly as moving,  $C'_s(\cdot) = C_m(\cdot)$ ,  $a$  is solution to the following equation:  $C'_s(r^*) = C_m(r^*)$ .

2. if sending is much more costly than moving,  $C'_s(\cdot) \gg C_m(\cdot)$ , then  $a$  is solution to the following equation:  $C'_s(r^*) = 10^2 C_m(r^*)$ .
3. if moving is much more costly than sending,  $C'_s(\cdot) \ll C_m(\cdot)$ , then  $a$  is solution to  $C'_s(r^*) = 10^{-2} C_m(r^*)$ .

**Energy consumed during consecutive routing.** In CoMNet as in MobileCOP, paths are computed for every packet transmission. As such, the first time a path is computed, nodes are more likely to move than in the second time a path is computed between the same source-destination pair. For this reason, we choose to evaluate the cumulative energy consumed during a fixed number of packet transmissions (or consecutive path computations) to allow a fair comparison of the schemes. Certain protocols do not need to move nodes again in the second route computation.

Figures 5(a), 5(c) and 5(e) display the cumulative energy spent (to move and to send) by each CoMNet variant and MobileCOP, for each mobility model for a node density of 300 nodes and as a function of the number of successive routings. Our extensive simulations show that the algorithms have the same behavior for low network node densities (due to page restrictions supporting data cannot be presented).

In  $Move_{(DSr)}$ , nodes are moved to the desired location every time a path is computed. However, in MobileCOP nodes are moved only after the first path is computed then-after, nodes memorize the path which will be used in future transmissions. Therefore, energy added at subsequent route computation results only from sending cost. Results show that *Routing on the move* and  $Move_{(r)}$  converge quickly since they use the same path after computing the first 3 consecutive routes from  $S$  to  $D$ . Since MobileCOP does not take into account the moving energy consumption in its routing decision, the distance moved by each node is the same for each model, but has different costs. Our analysis show that MobileCOP has a very high starting consumption cost due to node displacement. Since CoMNet incorporates the moving cost in its objective function, the initial cost remains low for CoMNet variants. Note that when moving is much more costly than sending (Figure 5(e)), *ORouting on the move* consumes the least energy, followed by  $Move_{(r)}$ . This is due to the inherent goals of each variant: *ORouting on the move* has a bias towards the move cost so it tries to minimize the move distance while  $Move_{(DSr)}$ 's priority is to minimize the sending cost.  $Move_{(r)}$  which aims at minimizing both energy costs has an average performance compared to the other variants. When sending cost is higher (Figure 5(c)) or close to moving cost (Figure 5(a)), the performance of all schemes are comparable. In this case, MobileCOP and  $Move_{(r)}$  have the best performance.

For all models, the cost difference between CoMNet and MobileCOP is even bigger for low node densities since nodes in MobileCOP move longer distances as candidate nodes are on average located farther away from the direct line between the source and the destination. This phenomena is counter-balanced in the other schemes by the fact that the lower the network density, the higher the proportion of dominating nodes. Thus, dominating nodes are more likely to be selected as next hop candidates, as a result routing will not incur any move cost.

**Effect of network density on protocol performance.** We select a source-destination pair ( $S$  and  $D$ ) and allow 10 consecutive route computations between  $S$  and  $D$ . Figures 5(b), 5(d) and 5(f) display the cumulative energy consumed after all consecutive routing tasks using each CoMNet variant and MobileCOP. The analysis is conducted for various network densities and for different mobility models. Note that MobileCOP runs independently of the mobility cost. The energy consumed smoothly decreases as the number of nodes increases. This is due to the fact that for high network density MobileCOP can select a relay node at the right location saving the energy to move it. To the contrary, CoMNet variants behave differently. When moving cost is equivalent to sending cost, CoMNet variants behave similar to MobileCOP. When moving is less costly than sending, CoMNet variants try to favor sending rather than moving. Nevertheless, for low densities, by construction of the CDS, almost every node belongs to the CDS and thus can not move. The energy consumed is thus mainly the energy consumed to send messages. Since sending is more expensive than moving, CoMNet consumes more energy than MobileCOP. As the network density increases, the proportion of dominant nodes decreases and thus, more nodes can move to the proper positions and hence, CoMNet variants outperform MobileCOP. For these settings, *ORouting* variant is the best performing technique (since it tries to minimize the moving distance), followed by  $Move_{(r)}$  and  $Move_{(DSr)}$ . The latter tries to favor sending over mobility energy consumption. When sending is more expensive than moving, CoMNet adapts to the network density (unlike MobileCOP where the moving cost is not considered in the optimized cost). For various network density, CoMNet outperforms MobileCOP. In addition, in all cases, CoMNet ensures the network connectivity and is memoryless, which is not the case for MobileCOP. For these settings, the best CoMNet variant is  $Move_{(DSr)}$  which tries to minimize the energy consumed to send messages.

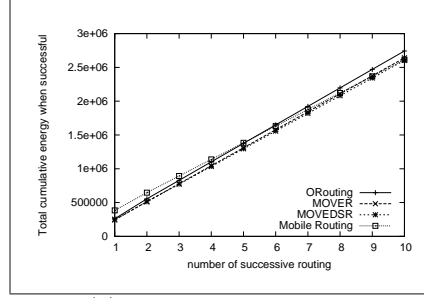
## 6 Conclusion and Future work

We introduce a novel protocol, CoMNet, that takes advantage of node mobility for efficient routing while ensuring network connectivity. The robustness of CoMNet compared to existing methods is due to the fact that it incorporates all costs (transmission as well as moving costs) in its routing decision. Note that the cost model does not have to be restricted to the transmission and moving costs it can be generalized to include receiving energy and other costs. Through extensive simulations, we show that CoMNet outperforms existing methods in terms of energy consumption and memory overhead. CoMNet has three variants that have different objective functions and mobility models. Each variant has its specific applications. Our future work will focus on the behavior of CoMNet in the presence of node conflict—when a node is solicited by more than one flow. Preliminary results show that  $Move_{(r)}$  is the most appropriate variant since it does not try to align nodes. Another interesting extension to this work would consider non-connected networks and explore node mobility to achieve network connectivity. This work is efficient for a single flow transmission, extending it to

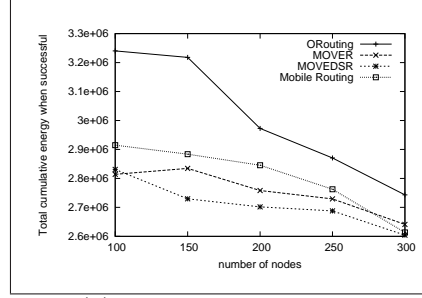
multiple flows may be an interesting problem; as flows share the same nodes and create bottlenecks, node conflict may complicate the routing schemes.

## References

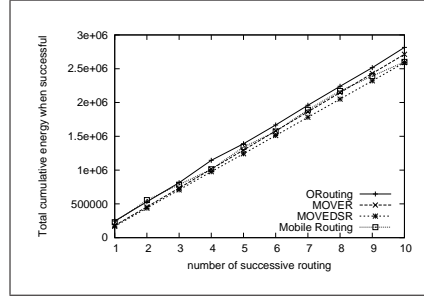
1. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad-hoc wireless networks. *ACM/Kluwer Wireless Networks*, 7(6):609–616, 2001.
2. J. Carle and D. Simplot-Ryl. Energy efficient area monitoring by sensor networks. *IEEE Computer Magazine*, 37:40–46, 2004.
3. F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Trans. Parallel and Distributed Systems (TPDS)*, 2004.
4. E. H. Elhafi, N. Mitton, and D. Simplot-Ryl. End-to-End Energy Efficient Geographic Path Discovery With Guaranteed Delivery in Ad hoc and Sensor Networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Cannes, France, September 2008.
5. E. Hamouda Elhafi and D. Simplot-Ryl. Flattening the gap between source-destination paths in energy efficient greedy georouting in wireless sensor networks. *Book Series Mobile Ad-Hoc and Sensor Networks.*, 4864:56–65, 2007.
6. G.G. Finn. Routing and addressing problems in large metropolitan-scale. *Inter-networks*, March 1987.
7. A. Fraboulet, G. Chelius, and E. Fleury. Worldsens: Development and prototyping tools for application specific wireless sensors networks. In *SPOTS*, April 2007.
8. H. Frey, S. Ruehrup, and I. Stojmenovic. Routing in wireless sensor networks. *Guide to Wireless Ad Hoc Networks*, 4:81–111, 2009.
9. A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic. Localized sensor area coverage with low communication overhead. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2006.
10. D.K. Goldenberg, J. Lin, and A.S. Morse. Towards mobility as a network control primitive. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 163–174, september 2004.
11. J. Kuruville, A. Nayak, and I. Stojmenovic. Progress and location based localized power aware routing for ad hoc sensor wireless networks. *Intern. Journal on Distributed Sensor Networks IJDSN*, 2:147–159, 2006.
12. H. Liu, A. Nayak, and I. Stojmenovic. Localized mobility control routing in robotic sensor wireless networks. *Mobile Ad-Hoc and Sensor Networks*, 4864:19–31, 2007.
13. V. Rodoplu and T. Meng. Minimizing energy mobile wireless networks. *IEEE Journal in Selected Areas in Communications JSAC*, 17(8):1333–1347, 1999.
14. J.A. Sanchez and P.M. Ruiz. Exploiting local knowledge to enhance energy-efficient geographic routing. In *International Conference on Mobile Sensor Networks (MSN)*, pages 567–578, December 2006.
15. D. Simplot-Ryl, I. Stojmenovic, and J. Wu. Energy efficient backbone construction, broadcasting, and area coverage in sensor networks. *Handbook of Sensor Networks: Algorithms and Architectures*, Wiley, pages 343–379, 2005.
16. I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Trans. Parallel and Distributed Systems (TPDS)*, 12(11):1122–1133, 2001.
17. W. Wang, V. Srinivasan, and K-C Chua. Extending the lifetime of wireless sensor networks through mobile relays. *IEEE/ACM Trans. Netw.*, 16(5):1108–1120, 2008.



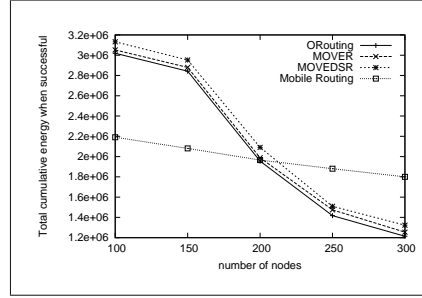
(a)  $C_s = C_m - 300$  nodes



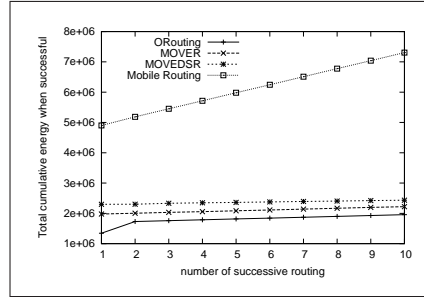
(b)  $C_s = C_m - 10$  routings



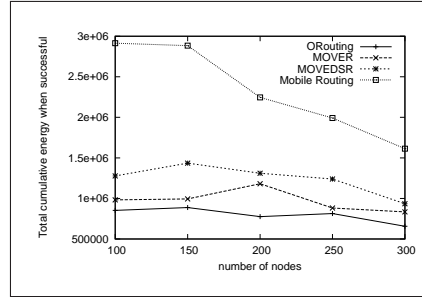
(c)  $C_s \gg C_m - 300$  nodes



(d)  $C_s \gg C_m - 10$  routings



(e)  $C_s \ll C_m - 300$  nodes



(f)  $C_s \ll C_m - 10$  routings

**Fig. 5.** Cumulative energy consumption for each algorithm.